Chapter 1
Summary of Session Control Tags

This chapter lists the tag elements that client applications and the JUNOScript server use to control the JUNOScript session. The tag names are in alphabetical order. For information about the notational conventions used in this chapter, see “Conventions for Tag Summaries” on page lxxviii.

<abort/>

Usage

```
<rpc>
  <any-child-of-rpc>
    <abort/>
  </any-child-of-rpc>
</rpc>
```

Description Direct the JUNOScript server to stop processing the request that is currently outstanding. The server responds by returning the <abort-acknowledgment/> tag, but might already have sent tagged data in response to the request. The client application must discard those tag elements.

See Also <abort-acknowledgment/> on page 3, <rpc> on page 17

<abort-acknowledgment/>

Usage

```
<rpc-reply>
  <any-child-of-rpc-reply>
    <abort-acknowledgment/>
  </any-child-of-rpc-reply>
</rpc-reply>
```

Description Indicate that the JUNOScript server has received the <abort/> tag and has stopped processing the current request. If the client application receives any tag elements related to the request between sending the <abort/> tag and receiving this tag, it must discard them.

See Also <xnm:error> on page 19, <rpc-reply> on page 18

<authentication-response>

Usage

```
<rpc-reply>
  <authentication-response>
      <status>authentication-outcome</status>
      <message>message</message>
  </authentication-response>
</rpc-reply>
```

Description Indicate whether an authentication attempt succeeded. The JUNOScript server returns the tag element in response to the <request-login> tag element emitted by a client application that uses the clear-text or Secure Sockets Layer (SSL) access protocol.

Contents message—Names the JUNOS account under which a connection to the JUNOScript server is established, if authentication succeeds. If authentication fails, explains the reason for the failure.

status—Indicates whether the authentication attempt succeeded. There are two possible values:

fail—The attempt failed. The JUNOScript server also emits the <challenge> tag element to request the password again, up to a maximum of three attempts.

success—The attempt succeeded. An authenticated connection to the JUNOScript server is established.

See Also <challenge> on page 4, <request-login> on page 16, <rpc-reply> on page 18

<challenge>

Usage

```
<rpc-reply>
  <challenge echo="no">Password:</challenge>
</rpc-reply>
```

Description Request the password associated with a JUNOS account during authentication with a client application that uses the clear-text or SSL access protocol. The JUNOScript server emits this tag element when the initial <request-login> tag element emitted by the client application does not enclose a <challenge-response> tag element, and when the password enclosed in a <challenge-response> tag element is incorrect (in the latter case, the server also emits an <authentication-response> tag element enclosing child tag elements that indicate the password is incorrect).

The tag element encloses the string Password: which the client application can forward to the screen as a prompt for a user.

Attributes echo—Specifies whether the password string typed by the user echoes on the screen. The value no specifies that it does not.

See Also <authentication-response> on page 4, <request-login> on page 16, <rpc-reply> on page 18

<command>

Usage

```
<rpc>
  <command>CLI-command-string</command>
</rpc>
```

Description Request that the JUNOScript server run the indicated JUNOS command-line interface (CLI) command. Provide the same options to the CLI command as when issuing the command at the CLI prompt.

Use this tag element only if the JUNOScript application programming interface (API) does not define an operational request tag element that corresponds to the CLI command. The output returned by the JUNOScript server in response to this tag element might be less complete or accurate than the output returned for a supported operational request tag element. For a list of the available operational request tag elements, see “Mapping between Operational Tags and CLI Commands” on page 23.

See Also <rpc> on page 17, “Mapping between Operational Tags and CLI Commands” on page 23

<commit-configuration>

Usage

```
<rpc>
  <commit-configuration/>

  <commit-configuration>
    <at-time>time-specification</at-time>
  </commit-configuration>

  <commit-configuration>
    <check/>
  </commit-configuration>

  <commit-configuration>
    <confirmed/>
    <confirm-timeout>rollback-delay</confirm-timeout>
  </commit-configuration>

  <commit-configuration>
    <synchronize/>
  </commit-configuration>

</rpc>
```

Description Request that the JUNOScript server perform one of the following actions, depending on the tag element's contents:

Commit the current candidate configuration, making it the active configuration on the router. Emit the empty <commit-configuration/> tag.

Schedule the current candidate configuration for commit at a future time. Enclose the <at-time> tag element in the <commit-configuration> tag element. There are three valid types of time specifiers:

The string reboot, to commit the configuration the next time the router reboots.

A time value of the form *hh*:*mm*[:*ss*] (hours, minutes, and optionally seconds), to commit the configuration at the specified time, which must be in the future but before 11:59:59 PM on the day the <commit-configuration> tag element is emitted. Use 24-hour time for the *hh* value; for example, 04:30:00 means 4:30:00 AM and 20:00 means 8:00 PM. The time is interpreted with respect to the clock and time zone settings on the router.

A date and time value of the form *yyyy-mm-dd hh*:*mm*[:*ss*] (year, month, date, hours, minutes, and optionally seconds), to commit the configuration at the specified date and time, which must be after the <commit-configuration> tag element is emitted. Use 24-hour time for the *hh* value. For example, 2003-03-21 12:30:00 means 12:30 PM on August 21, 2003. The time is interpreted with respect to the clock and time zone settings on the router.

The configuration is checked immediately for syntactic correctness. If the check succeeds, the configuration is scheduled for commit at the specified time. If the check fails, the commit operation is not scheduled.

Verify the syntactic correctness of the current candidate configuration without actually committing it. Enclose the <check/> tag in the <commit-configuration> tag element.

Commit the current candidate configuration but roll back to the previous configuration after a short time. Enclose the <confirmed/> tag in the <commit-configuration> tag element. By default, the rollback occurs after 10 minutes; to set a different rollback delay, also emit the optional <confirm-timeout> tag element.

To delay the rollback again (past the original rollback deadline), emit the <confirmed/> tag (enclosed in the <commit-configuration> tag element) again before the deadline passes. Include the <confirm-timeout> tag element to specify how long to delay the next rollback, or omit that tag element to use the default of 10 minutes. The rollback can be delayed repeatedly in this way.

To commit the configuration immediately and permanently after emitting the <confirmed/> tag, emit the empty <commit-configuration/> tag before the rollback deadline passes. The JUNOScript server commits the current candidate configuration and cancels the rollback. If the candidate configuration is still the same as the current committed configuration, this effectively recommits the current committed configuration.

On a router with multiple Routing Engines, copy the candidate configuration stored on the local Routing Engine to the other Routing Engine, verify the candidate's syntactic correctness, and commit it on both Routing Engines. Enclose the <synchronize/> tag in the <commit-configuration> tag element.

Contents at-time—Schedules the commit operation for a specified future time.

check—Requests verification that the configuration is syntactically correct, but does not actually commit it.

confirmed—Requests a commit of the current candidate configuration and a rollback to the previous configuration after a short time, 10 minutes by default. Use the <confirm-timeout> tag element to specify a different amount of time.

confirm-timeout—Specifies the number of minutes for which the configuration remains active when the <confirmed/> tag is enclosed in the <commit-configuration> tag element.

synchronize—Requests that the candidate configuration on the local Routing Engine be copied to the other Routing Engine, checked for correct syntax, and committed on both Routing Engines.

See Also <commit-results> on page 8, <rpc> on page 17

<commit-results>

Usage

```
<rpc-reply>
    <commit-results>
        <routing-engine>...</routing-engine>
    </commit-results>
</rpc-reply>
```

Description Enclose tag elements that contain information about a commit operation performed by the JUNOScript server on a particular Routing Engine.

Contents routing-engine—Information about a commit operation performed on a particular Routing Engine.

See Also <commit-configuration> on page 6, <routing-engine> on page 17, <rpc-reply> on page 18

<configuration>

Usage

```
<!-- when emitted by a client application -->
<rpc>
    <load-configuration>
        <configuration>
            <!-- tag elements representing configuration objects to load -->
        </configuration>
    </load-configuration>
</rpc>

<!-- when emitted by the JUNOScript server -->
<rpc-reply>
    <configuration>
        <!-- tag elements representing objects in the current configuration -->
    </configuration>
</rpc-reply>
```

Description Enclose JUNOScript tag elements that represent one or more configuration objects.

See Also <rpc> on page 17, <rpc-reply> on page 18

<configuration-text>

Usage

```
<!-- when emitted by a client application -->
<rpc>
  <load-configuration format=”text”>
    <configuration-text>
      <!-- ASCII text representing configuration objects to load -->
    </configuration-text>
  </load-configuration>
</rpc>

<!-- when emitted by the JUNOScript server -->
<rpc-reply>
  <configuration-text>
    <!-- ASCII text representing objects in the current configuration -->
  </configuration-text>
</rpc-reply>
```

Description Enclose ASCII text that represents one or more configuration objects.

See Also <rpc> on page 17, <rpc-reply> on page 18

<end-session/>

Usage

```
<rpc-reply>
  <end-session/>
</rpc-reply>
```

Description Indicate that the JUNOScript server is about to end the current session for a reason other than an error. Most often, the reason is that the client application has sent the <request-end-session/> tag.

See Also <request-end-session/> on page 16, <rpc-reply> on page 18

<get-configuration>

Usage

```
<rpc>
  <get-configuration [database="(candidate|committed)"] [format="(text|xml)"]
                  [inherit="inherit"] />

  <get-configuration [database="(candidate|committed)"] [format="(text|xml)"]]
                  [inherit="inherit"]>
    <!-- tag elements representing hierarchy level or object to display -->
  </get-configuration>
</rpc>
```

Description Request configuration data from the JUNOScript server. The database, format, and inherit attributes specify the source and formatting of the data to display. To display the entire hierarchy, emit the empty <get-configuration/> tag. To display one hierarchy level or a single configuration object, emit tag elements within the <get-configuration> tag element to represent all levels of the configuration hierarchy from the root (represented by the <configuration> tag element) down to the level or object to display. To represent the requested level, emit it as an empty tag. To represent the requested object, emit its container tag element and identifier tag element only, not any tag elements that represent other characteristics.

Attributes database—Specifies the configuration hierarchy from which to display data. There are two acceptable values:

candidate—The current candidate configuration

committed—The currently active (most recently committed) configuration on the router

format—Specifies the format in which the JUNOScript server returns the configuration data. There are two acceptable values:

text—Configuration statements are formatted as ASCII text, using the newline character, tabs and other white space, braces, and square brackets to indicate the hierarchical relationships between the statements. This is the format used in configuration files stored on the router and displayed by the JUNOS CLI show configuration command.

xml—Configuration statements are represented by the corresponding JUNOScript tag elements. This is the default value if the format attribute is omitted.

inherit—Specifies that the JUNOScript server enclose tag elements that are inherited from configuration groups within the tag elements that are inheriting them, and not display the <groups> tag element. The only acceptable value is inherit.

See Also <rpc> on page 17

<get-xnm-information>

Usage

```
<rpc>
  <get-xnm-information>
    <type>xml-schema</type>
    <namespace>junos-configuration</namespace>
  </get-xnm-information>
</rpc>
```

Description Request router configuration information represented as an Extensible Markup Language (XML) Schema. The JUNOScript server encloses the Schema in the <xsd:schema> tag element.

Contents namespace—Specifies the type of information for which to return an XML Schema. The only acceptable value is junos-configuration.

type—Specifies the format in which the JUNOScript server returns the configuration data. The only acceptable value is xml-schema.

See Also <rpc> on page 17

<junos:comment>

Usage

```
<rpc>
  <configuration>
    <!-- opening tags for one or more children of the <configuration> tag element -->
      <junos:comment>
        <!-- text of the comment-->
      </junos:comment>
      <!-- tag element representing configuration statement associated with the
               comment -->
    <!-- closing tags for one or more children of the <configuration> tag element -->
  </configuration>
</rpc>
```

Description

Enclose comment text to record with a configuration statement when the statement is added to the configuration database. The comment text string should not include any linebreak characters, but should include one of the two delimiters that indicate a comment in the configuration database: either the # character before the comment or the paired strings /* before the comment and */ after it.

The tag element that represents the configuration statement associated with the comment must appear directly after the closing </junos:comment> tag. If the associated tag element is omitted, the comment is not recorded in the configuration database.

See Also

<configuration> on page 8, <rpc> on page 17

<junoscript>

Usage

```
<!-- when emitted by a client application -->
<junoscript version="version" [hostname="hostname"] [release="release"]>
   <rpc>
      <!-- all tag elements generated by a client application -->
   </rpc>
</junoscript>

<!-- when emitted by the JUNOScript server -->
<junoscript version="version" hostname="hostname" os="os" release="release">
   xmlns="namespace-URL" xmlns:xnm="namespace-URL"
   xmlns:junos="namespace-URL"
   <rpc-reply>
      <!-- all tag elements generated by the JUNOScript server -->
   </rpc-reply>
</junoscript>
```

Description

Enclose all tag elements in a JUNOScript session (act as the root tag element for the session). The client application and JUNOScript server each emit this tag element, enclosing all other tag elements that they emit inside it.

Attributes

hostname—Names the machine on which the tag element's originator is running.

os—Specifies the operating system of the machine named by the hostname attribute.

release—Identifies the JUNOS release being run by the tag element's originator. JUNOS modules always set this attribute, but client applications do not have to.

version—(Required) Specifies the version of the JUNOScript API used for the enclosed set of tag elements.

xmlns—Names the XML namespace for the tag elements enclosed by the <junoscript> tag element that do not have a prefix on their names (that is, the default namespace for JUNOScript tag elements). The value is a URL of the form http://xml.juniper.net/xnm/*version-code*/xnm, where *version-code* is a string such as 1.1.

xmlns:junos—Names the XML namespace for the tag elements enclosed by the <junoscript> tag element that have the junos: prefix on their names. The value is a URL of the form http://xml.juniper.net/junos/*release-code*/junos, where *release-code* is the standard string that represents a release of the JUNOS software, such as 5.4R1 for the initial release of version 5.4.

xmlns:xnm—Names the XML namespace for the JUNOScript tag elements enclosed by the <junoscript> tag element that have the xnm: prefix on their names. The value is a URL of the form http://xml.juniper.net/xnm/*version-code*/xnm, where *version-code* is a string such as 1.1.

See Also

<rpc> on page 17, <rpc-reply> on page 18

<load-configuration>

Usage

```
<rpc>
  <load-configuration rollback="index" />

  <load-configuration url="url" [action="(merge|override|replace)"] \
                                                   [format="(text|xml)"] />

  <load-configuration format="text" [action="(merge|override|replace)"]>
    <configuration-text>
      <!-- formatted ASCII configuration statements to load -->
    </configuration-text>
  </load-configuration>

  <load-configuration [action="(merge|override|replace)"] [format="xml"]>
    <configuration>
      <!-- tag elements representing configuration statements to load -->
    </configuration>
  </load-configuration>
</rpc>
```

Description

Request that the JUNOScript server load configuration data into the current candidate configuration. Provide the data to load in one of three ways:

Set the empty <load-configuration/> tag's rollback attribute to the numerical index of a previous configuration. The router stores a copy of the most recently committed configuration and up to nine previous configurations, so valid values for the index are 0 (zero, for the most recently committed configuration) through 9 (for the oldest possible stored configuration). The previous configuration completely replaces the current configuration.

Set the empty <load-configuration/> tag's url attribute to the pathname of a file that resides on the router and contains the configuration data to load. The data can be either formatted ASCII (in which case the format attribute must be set to the value text) or JUNOScript tag elements (in which case the format attribute is either omitted or set to the value xml).

In the following example, the url attribute identifies /tmp/add.conf as the file to load. (The omission of the format attribute indicates that the file contains JUNOScript tag elements.)

```
<load-configuration url="/tmp/add.conf"/>
```

Enclose the configuration data within an opening <load-configuration> and closing </load-configuration> tag. If providing the configuration data as formatted ASCII, enclose it in a <configuration-text> tag element and set the format attribute to the value text. If providing configuration data as JUNOScript tag elements, enclose it in a <configuration> tag element and optionally set the format attribute to the value xml.

Attributes

action—Specifies how to load the configuration data, particularly when the current candidate contains conflicting statements. There are three acceptable values:

merge—Combines the data in the loaded configuration with the current candidate configuration. If statements in the loaded configuration conflict with statements in the current candidate configuration, the loaded statements replace the current ones. This is the default behavior if the action attribute is omitted.

override—Discards the entire current candidate configuration and replaces it with the loaded configuration.

replace—Substitutes each hierarchy level or configuration object defined in the loaded configuration for the corresponding level or object in the current configuration.

If providing the configuration data as formatted ASCII (either in the file named by the url attribute or enclosed in a <configuration-text> tag element), also place the replace: statement on the line directly preceding the statements that represent the hierarchy level or object to replace. For more information, see the section about loading a configuration in the *JUNOS Internet Software Configuration Guide: Getting Started*.

If providing the configuration data as JUNOScript tag elements, also set the replace attribute to the value replace on the opening tag of the container tag element that represents the hierarchy level or object to replace.

format—Specifies the format used for the configuration data. There are two acceptable values:

text—Indicates that configuration statements are formatted as ASCII text, using the newline character, tabs and other white space, braces, and square brackets to indicate the hierarchical relationships between the statements. This is the format used in configuration files stored on the router and displayed by the JUNOS CLI show configuration command.

xml—Indicates that configuration statements are represented by the corresponding JUNOScript tag elements. This is the default value if the format attribute is omitted.

rollback—Specifies the numerical index of the previous configuration to load. Valid values for the index are 0 (zero, for the most recently committed configuration) through 9 (for the oldest possible stored configuration).

url—Specifies the full pathname of the file that contains the configuration data to load. The file must reside on the router's local disk.

See Also

<configuration> on page 8, <configuration-text> on page 9, <load-configuration-results> on page 15, <rpc> on page 17

<load-configuration-results>

Usage

```
<rpc-reply>
  <load-configuration-results>
    <load-success/>
    <load-error-count>errors</load-error-count>
  </load-configuration-results>
</rpc-reply>
```

Description Enclose one of the two following tag elements, which indicate the status of a configuration loading operation performed by the JUNOScript server.

Contents load-error-count—Specifies the number of errors that occurred when the JUNOScript server attempted to load new data into the candidate configuration. The candidate configuration must be restored to a valid state before it is committed.

load-success—Indicates that the JUNOScript server successfully loaded new data into the candidate configuration.

See Also <load-configuration> on page 13, <rpc-reply> on page 18

<lock-configuration>

Usage

```
<rpc>
  <lock-configuration/>

  <lock-configuration>
    <rollback>automatic</rollback>
  </lock-configuration>
</rpc>
```

Description Request that the JUNOScript server open and lock the current candidate configuration, enabling the client application both to read and change it, but preventing any other users or applications from changing it. The application must emit the <unlock-configuration/> tag to unlock the configuration.

To specify that any uncommitted changes are discarded from the candidate configuration if the JUNOScript session ends before the client application commits the configuration, set the value of the <rollback> tag element to automatic and enclose the <rollback> tag element in the <lock-configuration> tag element.

See Also <rpc> on page 17, <unlock-configuration/> on page 19

<output>

Usage

```
<rpc-reply>
  <output>response</output>
</rpc-reply>
```

Description Contain a response to a client application request when the current version of the JUNOScript API does not define a specific tag element for the response. The format of the contents is subject to change, so the client application must not rely on a particular format.

See Also <rpc-reply> on page 18

<request-end-session/>

Usage

```
<rpc>
    <request-end-session/>
</rpc>
```

Description Request that the JUNOScript server end the current session.

See Also <end-session/> on page 9, <rpc> on page 17

<request-login>

Usage

```
<rpc>
    <request-login>
        <username>JUNOS-account</username>
        <challenge-response>password</challenge-response>
    </request-login>
</rpc>
```

Description Request authentication by the JUNOScript server when using the clear-text or SSL access protocol.

Emitting both the <username> and <challenge-response> tag elements is appropriate if the client application automates access to router information and does not interact with users, or obtains the password from a user before beginning the authentication process.

Emitting only the <username> tag element is appropriate if the application does not obtain the password until the authentication process has already begun. In this case, the JUNOScript server returns the <challenge> tag element to request the password associated with the account.

Contents challenge-response—Specifies the password for the JUNOS account named in the <username> tag element. Omit this tag element to have the JUNOScript server emit the <challenge> tag element to request the password.

username—Names the JUNOS account under which to authenticate with the JUNOScript server. The account must already be configured on the router where the JUNOScript server is running.

See Also <challenge> on page 4, <rpc> on page 17

<routing-engine>

Usage

```
<rpc-reply>
  <commit-results>

<!-- to indicate successful commit of the candidate configuration -->
    <routing-engine>
      <name>reX</name>
      <commit-success/>
    </routing-engine>

<!-- to indicate that candidate configuration is syntactically valid -->
    <routing-engine>
      <name>reX</name>
      <commit-check-success/>
    </routing-engine>

  </commit-results>
</rpc-reply>
```

Description Enclose tag elements indicating that the JUNOScript server successfully fulfilled a commit request.

Contents commit-check-success—Indicates that the candidate configuration is syntactically correct.

commit-success—Indicates that the JUNOScript server successfully committed the candidate configuration.

name—Name of the Routing Engine on which the commit operation was performed. Possible values are re0 and re1.

See Also <commit-results> on page 8, <rpc-reply> on page 18

<rpc>

Usage

```
<junoscript>
  <rpc>
    <!-- all other tag elements generated by a client application -->
  </rpc>
</junoscript>
```

Description Enclose all tag elements generated by a client application.

See Also <junoscript> on page 12, <rpc-reply> on page 18

<rpc-reply>

Usage

```
<junoscript>
  <rpc-reply xmlns:junos="namespace-URL">
    <!-- all tag elements generated by the JUNOScript server -->
  </rpc-reply>
</junoscript>
```

Description Enclose all tag elements generated by the JUNOScript server. The immediate child tag element is usually one of the following:

The specific tag element used to enclose data generated by the module in response to a client application’s request.

The <output> tag element, if the JUNOScript API does not define a specific tag element for the type of information in the response.

Attributes xmlns:junos—Names the XML namespace for the JUNOScript tag elements enclosed by the <rpc-reply> tag element that have the junos: prefix on their names. The value is a URL of the form http://xml.juniper.net/junos/*release-code*/junos, where *release-code* is the standard string that represents a release of the JUNOS Internet software, such as 5.3R1 for the initial release of version 5.3.

See Also <junoscript> on page 12, <output> on page 15, <rpc> on page 17

<undocumented>

Usage

```
<rpc-reply>
  <undocumented>
    <!-- tag element representing unsupported configuration element -->
  </undocumented>
</rpc-reply>
```

Description Enclose a tag element representing a configuration element (hierarchy level or object) that is not documented in the JUNOS software configuration guides or officially supported by Juniper Networks. The reason that the element is undocumented is usually one of the following:

It is used for debugging purposes only by Juniper Networks personnel.

It is no longer supported or has been moved to another area of the configuration hierarchy, but appears in the current location for backward compatibility.

See Also <rpc-reply> on page 18

<unlock-configuration/>

Usage

```
<rpc>
   <unlock-configuration/>
</rpc>
```

Description Request that the JUNOScript server unlock and close the current candidate configuration. Until the application emits this tag, other users or applications can read the configuration but cannot change it.

See Also <lock-configuration> on page 15, <rpc> on page 17

<xnm:error>

Usage

```
<junoscript>
   <any-child-of-junoscript>
      <xnm:error xmlns="namespace-URL" xmlns:xnm="namespace-URL">
         <parse/>
         <source-daemon>module-name</source-daemon>
         <filename>filename</filename>
         <line-number>line-number</line-number>
         <column>column-number</column>
         <token>input-token-id</token>
         <message>error-string</message>
         <edit-path>edit-path-name </edit-path>
      </xnm:error>
   </any-child-of-junoscript>
</junoscript>
```

Description Indicate that the JUNOScript server has experienced an error while processing the client application's request. If the server has already emitted the response tag element for the current request, the information enclosed in the response tag element might be incomplete. The client application must include code that discards or retains the information, as appropriate. The child tag elements described in the Contents section detail the nature of the error. The JUNOScript server does not necessarily emit all child tag elements; it omits tag elements not relevant to the current request.

Attributes xmlns—Names the XML namespace for the contents of the tag element. The value is a URL of the form http://xml.juniper.net/xnm/*version*/xnm, where *version* is a string such as 1.1.

xmlns:xnm—Names the XML namespace for child tag elements that have the xnm: prefix on their names. The value is a URL of the form http://xml.juniper.net/xnm/*version*/xnm, where *version* is a string such as 1.1.

Contents <column>—Identifies the element that caused the error by specifying its position as the number of characters after the first character in one of the following:

The JUNOS CLI command string enclosed in a <command> tag element sent by the client application and currently being processed

The line specified by the <line-number> tag element in the configuration file that was being loaded (which is named in the <filename> tag element)

<edit-path>—Specifies the JUNOS CLI configuration-mode edit path in effect when the error occurred (provided only during loading of a configuration file).

<filename>—Names the configuration file that was being loaded.

<line-number>—Specifies the line number where the error occurred in the configuration file that was being loaded, which is named by the <filename> tag element.

<message>—Describes the error in a natural-language text string.

<parse/>—Indicates that there was a syntactic error in the request submitted by the client application.

<source-daemon>—Names the JUNOS module that was processing the request in which the error occurred.

<token>—Names which element in the request caused the error.

See Also <junoscript> on page 12, <xnm:warning> on page 20

<xnm:warning>

Usage

```
<junoscript>
  <any-child-of-junoscript>
    <xnm:warning xmlns="namespace-URL" xmlns:xnm="namespace-URL">
      <source-daemon>module-name</source-daemon>
      <filename>filename</filename>
      <line-number>line-number</line-number>
      <column>column-number</column>
      <token>input-token-id</token>
      <message>error-string</message>
    </xnm:warning>
  </any-child-of-junoscript>
</junoscript>
```

Description Indicate that the server has encountered a problem while processing the client application's request. The child tag elements described in the Contents section detail the nature of the warning.

Attributes xmlns—Names the XML namespace for the contents of the tag element. The value is a URL of the form http://xml.juniper.net/xnm/*version*/xnm, where *version* is a string such as 1.1.

xmlns:xnm—Names the XML namespace for child tag elements that have the xnm: prefix on their names. The value is a URL of the form http://xml.juniper.net/xnm/*version*/xnm, where *version* is a string such as 1.1.

Contents <column>—Identifies the element that caused the error by specifying its position as the number of characters after the first character in one of the following:

The JUNOS CLI command string enclosed in a <command> tag element sent by the client application and currently being processed

The line specified by the <line-number> tag element in the configuration file that was being loaded (which is named in the <filename> tag element)

<filename>—Names the configuration file that was being loaded.

<line-number>—Specifies the line number where the error occurred in the configuration file that was being loaded, which is named by the <filename> tag element.

<message>—Describes the warning in a natural-language text string.

<source-daemon>—Names the JUNOS module that was processing the request in which the warning occurred.

<token>—Names which element in the request caused the warning.

See Also <xnm:error> on page 19, <junoscript> on page 12